UCRL-ID-151789

Numerical Technology for Large-Scale Computational Electromagnetics

R. M. Sharpe, N. J. Champagne, D. A. White, M. Stowell, and R. Adams

January 30, 2003

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

Numerical Technology for Large-Scale Computational Electromagnetics

Robert M. Sharpe, Ph.D.
Electronics Engineering

Nathan J. Champagne, Ph.D.
Electronics Engineering

Daniel A. White, Ph.D.
Center for Applied Scientific Computing

Mark Stowell
Center for Applied Scientific Computing

Rob Adams, Ph.D.
University of Kentucky

Abstract

The key bottleneck of implicit computational electromagnetics tools for large complex geometries is the solution of the resulting linear system of equations. The goal of this effort was to research and develop critical numerical technology that alleviates this bottleneck for large-scale computational electromagnetics (CEM). The mathematical operators and numerical formulations used in this arena of CEM yield linear equations that are complex valued, unstructured, and indefinite. Also, simultaneously applying multiple mathematical modeling formulations to different portions of a complex problem (*hybrid* formulations) results in a mixed structure linear system, further increasing the computational difficulty. Typically, these hybrid linear systems are solved using a direct solution method, which was acceptable for Cray-class machines but does not scale adequately for ASCI-class machines. Additionally, LLNL's previously existing linear solvers were not well suited for the linear systems that are created by hybrid implicit CEM codes. Hence, a new approach was required to make effective use of ASCI-class computing platforms and to enable the next generation design capabilities. Multiple approaches were investigated, including the latest sparse-direct methods developed by our ASCI collaborators. In addition, approaches that combine domain decomposition (or matrix partitioning) with general-purpose iterative methods and special purpose pre-conditioners were investigated. Special-purpose pre-conditioners that take advantage of the structure of the matrix were adapted and developed based on intimate knowledge of the matrix properties. Finally, new operator formulations were developed that radically improve the conditioning of the resulting linear systems thus greatly reducing solution time. The goal was to enable the solution of CEM problems that are 10 to 100 times larger than our previous capability.

Introduction

The key bottleneck of implicit computational electromagnetics tools for large complex geometries is the solution of the resulting linear system of equations. The purpose of this LDRD effort was to develop critical numerical technology that alleviates this bottleneck for large-scale computational electromagnetics (CEM). By large-scale CEM, we mean high-fidelity direct numerical solution of Maxwell's equations on ASCI class massively parallel computers (teraFLOPS speed, teraBYTES memory).

A prime example of the present CEM class of tools is the Electromagnetic Interactions GEneRalized (EIGER) code, which was developed within Electronics Engineering. The EIGER code is a state-of-the-art frequency domain Maxwell equation solver based on a hybrid Finite Element – Boundary Element (FE-BE) formulation [1]. The code has an object-oriented design that has been implemented in Fortran 90 and allows a wide variety of numerical formulations, operators, and analytical treatments to be applied to the same problem simultaneously. Although this represents a real breakthrough in our ability to concisely apply different physics and numerics to a complex situation, it manifests itself mathematically in a mixed structure linear system that is extremely difficult to solve efficiently. This class of implicit codes (EIGER, NEC, and PATCH) represents some of Engineering's general-purpose "workhorse" CEM tools. They have been used for modeling structures as diverse as MEMs devices and EUV lithography applications to the installed performance of full-scale antenna systems on complex platforms (such as NAI satellites and Navy surface combatants). An ability to solve larger systems of equations would enable modeling of more complex geometries and physically larger problems at a finer level of detail.

Since the linear solve process is the computational bottleneck of implicit codes, we developed parallel linear solver technology for scientific and engineering codes that solve large, complex, unstructured, indefinite linear systems using EIGER as a test bed. Previously, the EIGER code solved a large linear system of equations using a direct solution method with $O(N^3)$ complexity. Note that this is in contrast to many other Laboratory codes that are completely explicit (solving no linear system of equations) and therefore straightforward to parallelize. While the direct method was acceptable for Cray-class machines, it is unacceptable for ASCI-class machines. We estimate that a medium sized problem with 200000 unknowns would require 30 CPU days on our ASCI Blue supercomputer! LLNL has a number of efficient linear solvers, but these solvers are not suited for the linear systems addressed in this project. As discussed later, the mixed structure and mathematical nature of CEM problems present unique challenges that had not been adequately addressed by the applied math community. Thus, we required a new approach in order to make effective use of these world-class computing platforms and to enable the next generation design capabilities. Our goal was to enable ASCI-class machines to solve CEM problems that are 10 to 100 times larger than our previous capability. This would enable an unprecedented design capability with improved fidelity and accuracy. In the sections that follow, we describe our novel approaches for this problem and summarize our progress and accomplishments.

Background

Implicit CEM tools, such as EIGER, are currently used to support a wide variety of LLNL programs and work for others (WFO) efforts. Each of these activities is constrained by the electrical size or modeling detail of the problem which then manifest themselves in the corresponding memory requirements and speed limitations of present solver technology. These modeling activities include:

- NAI - Analysis and design of broadband RF systems for data collection and communication within the intelligence community
- Lasers (IS&T - EUV Lithography) – Three-dimensional full-wave analysis of defects and the resulting effects on imaging quality and printability
- Earth and Environmental – Large and rapid forward EM simulations coupled to inversion algorithms for underground imaging
- Lasers and D&NT (MEMs) – Accurate prediction of electrical forces on micro-machines coupled to mechanical simulations for deformation analysis and component performance. Applications include surety components for weapons and micro-mirrors for IS&T
- D&NT – Analysis of high powered pulsed electromagnetic systems
- DoD – High Performance Computing Modernization Office (HPCMO) investment in massively parallel predictive tools for the war fighter program. In addition, Navy systems design and topside installed performance on next generation ships.
- DARPA – Mixed signal systems CAD program (started July ’01) to couple full-wave EM solvers with digital and analog circuit simulators to model the next generation of systems-on-a-chip.

Related Activities

Linear solver technology is an essential element for most of the disciplines, tools, and applications in the Center for Computational Engineering (CCE). Computational engineering and the CCE at LLNL, in particular, needed a library of parallel linear solvers to address the needs of the laboratory programs and to effectively utilize the present revolution in computer hardware.

Computational electromagnetics is a core technology of the laboratory, with electromagnetic design and analysis directly impacting up to 50% of EE’s business. The CEM tools are classified into two broad areas: frequency domain methods and time domain methods. All of the frequency domain methods result in an underlying linear system of equations to be solved. The time domain methods are being extended to include implicit methods that will also require the solution of large linear systems by FY00. Improved solvers are therefore an enabling technology for virtually all of the EM tools within the CCE that are targeted for high performance computer (HPC) platforms.

Significant research and development activities were leveraged during the course of this project. Although the ASCI linear solver developments are not focused on CEM related issues, key researchers there were interested in extending their research to include our problem domain.

Also, the work performed for sparse iterative solvers in the CEM arena has paved the way for extensions to the ME implicit tools (like NIKE) which currently rely solely on parallel direct methods. This was facilitated by coordinating our efforts with the Finite Element Interface (FEI), and ongoing ASCI activity to standardize the interface between solvers and applications. In addition, CASC has an entire group that is responsible for linear solver technology focused primarily on the solution of elliptic partial differential equations. These researchers collaborated to address the difficult and unique challenges presented by this class of problem.

Approach

The EIGER program solves the frequency domain Maxwell's Equations for arbitrary 3D geometry's involving conductors, dielectric, and magnetic media. The user can specify the known fields and currents, and EIGER computes the unknown fields and currents in a completely self-consistent and physically accurate manner. In EIGER, Maxwell's equations are solved using a hybrid Partial Differential Equation – Integral Equation (PDE-IE) formulation. The PDE part of the formulation is the exact physical description of the electromagnetic fields and currents in the inhomogeneous interior of a problem, whereas the IE part of the formulation is the exact physical description of the propagation of waves in the "unbounded" region. The integral equation portion of the problem can also employ analytical treatments by using Green's functions to satisfy many boundary conditions directly (i.e. layered media and periodic structures). This hybrid formulation is optimal for a wide variety of open region EM problems such as antenna design, radar cross section prediction, and optical scattering since the BE formulation exactly models the radiation of EM fields in the infinite domain.

The natural discretization of the PDE-IE formulation used in EIGER is a hybrid Finite Element – Boundary Element (FE-BE) discretization, where the interior PDE is discretized using volumetric finite elements and the exterior is discretized using surface boundary elements. This results in a large linear system of equations that must be solved for the unknown fields and currents. The solution of this linear system is the computational bottleneck and was the emphasis of this LDRD effort. The linear system is complex, non-Hermitian, indefinite, and ill conditioned. In addition, the linear system is complex because we are solving the frequency domain Maxwell's Equations. The linear system is non-Hermitian since the PDE-IE itself is not self-adjoint; this in contrast to other fields of physics where the PDE is self-adjoint. Our linear system is indefinite due to the nature of our Helmholtz operator. This is quite different than the linear systems generated by the more abundant solid mechanics, hydrodynamics, and radiation transport codes; hence, the applied mathematics community has not aggressively attacked our problem. Finally, our linear system is ill conditioned because the underlying integro-differential equation we are discretizing is extremely stiff.

EIGER previously used the ScaLAPACK library to solve the linear system [3]. This is a general-purpose parallel dense matrix library. Since it is based on dense matrices, the CPU time for a solution is $O(N^3)$, where N is the size of the matrix. For small systems ($N < 10{,}000$) this does not present a major problem. However, in order to obtain a high fidelity solution (using a fine computational mesh) or to address electrically larger problems a much larger matrix is required. As stated in section 2, a "medium" sized problem with $N = 200000$ would require 30 days of CPU time on the ASCI Blue supercomputer (figure 1).

Our approach to overcoming the $O(N^3)$ bottleneck was to combine general-purpose iterative methods with special purpose preconditioners that take advantage of the structure of the matrix. In an iterative method, the user supplies an initial starting vector $\mathbf{x^0}$, and each iteration provides a new vector $\mathbf{x^i}$, which is closer to the exact solution **x**. There is an important class of iterative methods that require no matrix operations other than a simple matrix vector multiplication. In these methods, the *i*th iterate $\mathbf{x^i}$ is an element of the space $\{\mathbf{r^0}, \mathbf{A\ r^0},...,\mathbf{A^i\ r^0}\}$, where $\mathbf{r^0 = A\ x - b}$. This space is known as a Krylov space, and these methods are called Krylov methods. The most well known Krylov method is of course the Conjugate Gradient (CG) method, which is applicable to symmetric positive definite matrices. More modern Krylov methods applicable to our non-symmetric indefinite systems include Quasi Minimum Residual (QMR) and General Minimum Residual (GMRES).

Each iteration of a Krylov method requires multiplication of the residual by the matrix **A**. If **A** is sparse, each iteration is $O(N)$, where N is the size of the matrix. If A is dense, the complexity is $O(N^2)$. In either case, if only a few iterations are required, a Krylov method can be significantly faster than a direct method.

Accurate predictions of the rate of convergence of Krylov methods are difficult to make, but there are some general results. For example, the convergence of the CG method for symmetric positive definite systems is known to be proportional to the square root of the condition number of the matrix (where the condition number of such a matrix is the ratio of the maximum eigenvalue to the minimum eigenvalue). Hence, for well-conditioned systems CG converges quite rapidly. For other Krylov methods such GMRES, the clustering of the eigenvalues has a significant impact on convergence.

In practice, preconditioning of the system of equations is often employed to improve the rate of convergence. The idea is to apply a linear transformation to the system of equations such that the transformed system is better conditioned, and then the Krylov method is applied to the transformed system and should converge faster. For example, consider the linear system $\mathbf{A\ x = b}$. We can multiply both sides of the equation by a matrix $\mathbf{M^{-1}}$ without affecting the solution, resulting in the new system $\mathbf{M^{-1}\ A\ x = M^{-1}\ b}$. The matrix **M** is referred to as the preconditioner. In order for this to be an effective preconditioner, 1) the new matrix $\mathbf{M^{-1}\ A}$ must be better conditioned than the original matrix **A**, resulting in fewer Krylov iterations, and 2) the matrix $\mathbf{M^{-1}}$ must be easy (i.e. computationally inexpensive) to compute. Generally, if **M** is in some sense close to **A**, then $\mathbf{M^{-1}\ A}$ is close to the identity matrix (ideally conditioned) and the Krylov method will converge in a few iterations.

The most effective preconditioners are those that take advantage of knowledge of the special properties of the underlying operator. For example, consider the solution of Laplace’s equation using a finite difference or finite element method. Since the Laplacian is known to be a smoothing operator, using a coarse grid solution, as a preconditioner is extremely effective, this is in fact the basis for CASC’s multigrid effort [4]. As another example, in radiation transport the diffusion approximation can be used to precondition the full transport equation. Unfortunately, neither of these approaches is applicable to Maxwell’s Equations.

Our approach was to develop special-purpose preconditioners based on intimate knowledge of the special properties of our PDE-IE operator. For example, we know that our hybrid PDE-IE formulation results in a block-structured matrix where we have a large sparse block due to the PDE and a smaller dense block due to the IE. This is illustrated in Figure 2.

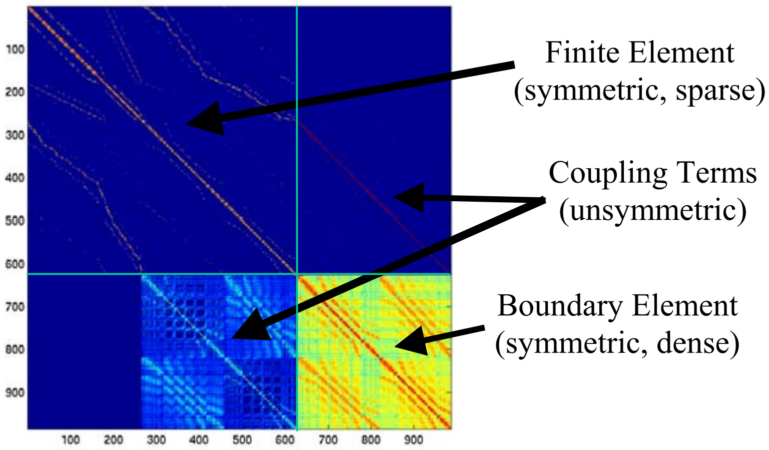


Figure 2. Our hybrid PDE-IE formulation results in a block-structured matrix consisting of a large, sparse block and a smaller dense block. We can design efficient parallel preconditioners that take advantage of this structure.

We can write our matrix **A** in block form where **S** is the large sparse block, **D** is the small dense block, and the matrices **C** and **Q** are the coupling terms (figure 3). This is in effect a domain decomposition where our unknowns are decomposed into volume (interior) and surface (boundary) unknowns, denoted as $\mathbf{X_v}$ and $\mathbf{X_s}$, respectively. Likewise, the unknown source terms are decomposed into volume and surface terms, denoted as $\mathbf{b_v}$ and $\mathbf{b_s}$, respectively.

$$\left[\begin{array}{c:c} \mathbf{S} & \mathbf{C} \\ \hdashline \mathbf{Q} & \mathbf{D} \end{array}\right] \left[\begin{array}{c} \mathbf{X_v} \\ \hdashline \mathbf{X_s} \end{array}\right] = \left[\begin{array}{c} \mathbf{b_v} \\ \hdashline \mathbf{b_s} \end{array}\right]$$

Figure 3. Decomposing the unknowns into volume and surface domains leads to a block structured matrix for which we can easily design preconditioners.

Several preconditioning methods become apparent now that our matrix is in block form. For example, we can construct a preconditioner **M** by ignoring the coupling terms in **A**, as illustrated in Figure 4. It is important to remember that **M** is only a preconditioner; we are solving the exact system **A x = b** using a Krylov method. The purpose of the preconditioner **M** is to reduce the number of Krylov iterations. To implement this in parallel, we assign *m* processors to work on $\mathbf{S}^{-1}$ and *n* processors to work on $\mathbf{D}^{-1}$. Since **D** is dense, we initially used standard dense solver technology to solve this portion. As **D** increases in size, an iterative approach can be applied to its solution. This methodology is in contrast to our previous approach of using ScaLAPACK on the entire system, which as described earlier is unacceptable for ASCI-scale EM problems.

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{S} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{D} \end{array}\right]$$

Figure 4. A preconditioner **M** constructed by ignoring the coupling terms in **A**. A group of m processors is assigned to work on $\mathbf{S}^{-1}$, another group of n processors is assigned to work on $\mathbf{D}^{-1}$.

Several approaches exist for computing $\mathbf{S}^{-1}$. Since **S** is extremely sparse, a sparse direct method based on degree-of-freedom re-ordering can be employed. Sparse direct methods were developed specifically for solving sparse systems of equations resulting from finite difference for finite element methods. By solving, we mean computing the lower and upper triangular factors of the matrix. To be specific, if our original system of equations is $\mathbf{S}\ \mathbf{x} = \mathbf{b}$, then we perform the decomposition $\mathbf{LU} = \mathbf{S}$ and subsequently solve the two triangular systems $\mathbf{L}\ \mathbf{z} = \mathbf{b}$ and $\mathbf{U}\ \mathbf{x} = \mathbf{z}$. The calculation of the triangular matrices **L** and **U** is the computationally intensive part of the process, and for general dense matrices is $O(N^3)$ where N is the size of the matrix.

Original System	$\mathbf{S}\ \mathbf{x} = \mathbf{b}$
1.) Decomposition	$\mathbf{LU} = \mathbf{S}$
2.) Forward Substitution	$\mathbf{z} = \mathbf{L}^{-1}\ \mathbf{b}$
3.) Backward Substitution	$\mathbf{x} = \mathbf{U}^{-1}\ \mathbf{z}$

Our matrix **S** is assumed to be sparse and unstructured. Unfortunately, the matrices **L** and **U** may not be as sparse as the original matrix **S**. The sparcity of **L** and **U**, and hence the computational complexity of the problem, is profoundly dependent upon the ordering of the original matrix **S**. Re-ordering the unknowns to reduce the size of the **L** and **U** matrices has a directed impact on the parallel solution of the system. This is extremely important for parallel systems such as the ASCI Blue machine, since it minimizes the amount of message passing. Without reordering, it is possible that every processor communicates data with every other processor, creating a show-stopping bottleneck in the computation. With optimal re-ordering only nearest-neighbor communication is required.

Several types of orderings such as multilevel red-black can be used to achieve a higher degree of scalability [5]. These orderings attempt to redistribute the unknowns in a manner that reduces serialism, with an increase in communication efficiency. Thus, the optimal ordering is often dependent upon the specifics of the parallel system. Using reordering of the unknowns, sparse direct methods have been shown to have $O(N^2)$ complexity for a wide variety of finite difference and finite element applications [6]. Typically, if the matrix **A** has m non-zero entries per row, then the matrices **L** and **U** will have m^2 non-zero entries. For large systems of equations $m^2 << N$ and we have a significant computational savings. These methods had not previously been applied to electromagnetic problems, and these methods have not been tried on ASCI class machines.

The approach described above is only one of the approaches pursued for solving this difficult problem. Other approaches investigated include Shur complement methods which first solve for the boundary unknowns $\mathbf{X_s}$ using the Shur complement of **A**, and then solve for the interior unknowns $\mathbf{X}_v$ [5]. In a parallel implementation all *n* processors work on computing $\mathbf{X}_s$**,** which involves dense matrix linear algebra, and then all *n* processors work on computing $\mathbf{X}_v$**,** which involves only sparse matrix algebra. This approach also takes into account the structure of the matrix, but parallelism is accomplished in a different way.

Other viable approaches investigated for solving our system of equations included sparse approximate inverse (SPAI) techniques, incomplete LU factorization (ILU), and fast multi-pole methods (FMM). The SPAI technique involves computing a preconditioner **M** that is a sparse approximate inverse of the original matrix **A**, hence application of the preconditioner is trivial to parallelize. The primary research issue here is computing an effective approximate inverse [7]. Fast multi-pole methods employ a multipole expansion of the EM fields; an exact expansion can be used as an alternative to storing the dense matrix **D**, or a truncated expansion can be used an a preconditioner to the matrix **D**. The FMM has been shown to be an $N \log_2 N$ method for a certain class of simple CEM problems [8], the research issue here was to determine how effective it is for our inhomogeneous problems.

Progress and Results

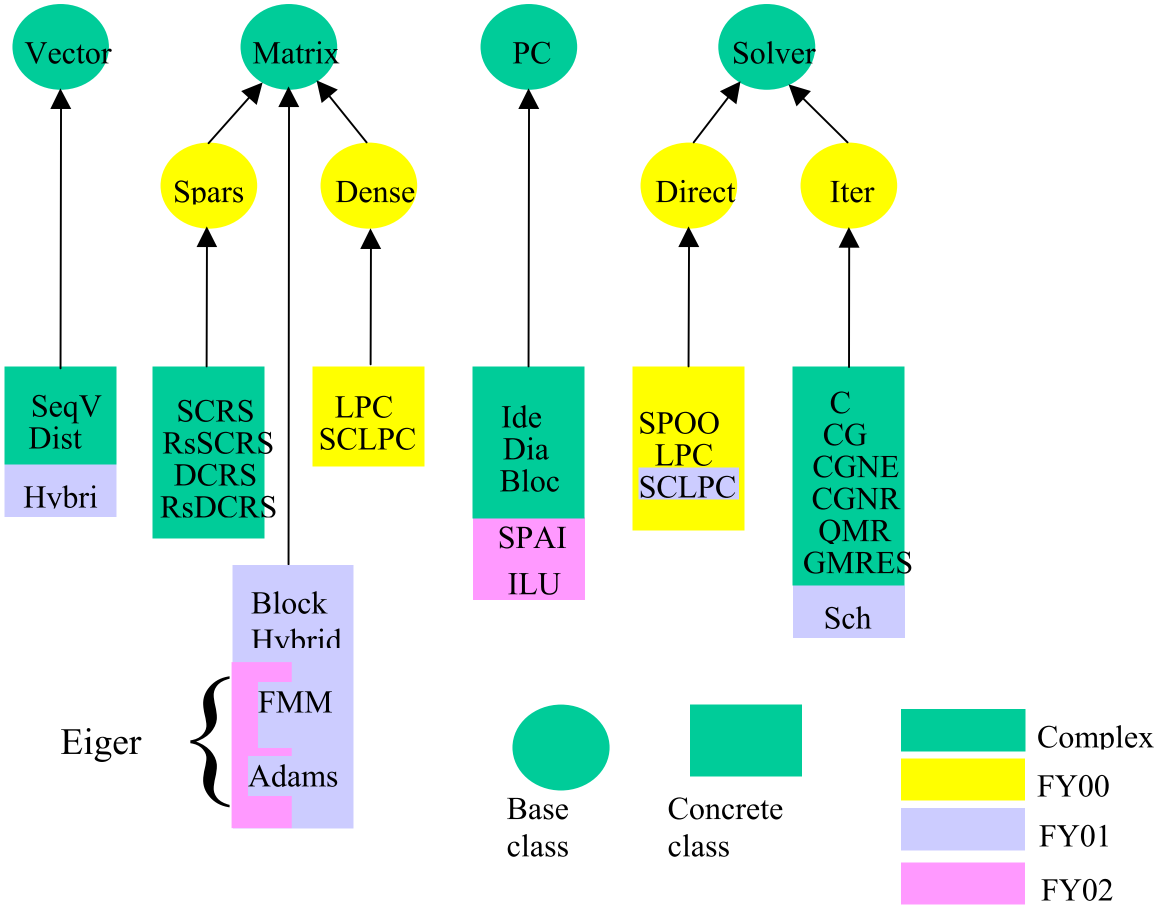
Collection of Representative Matrices

The first activity completed under this effort, was the development of a CEM matrix test-bed by collecting a variety of linear systems of equations. These systems were comprised of various physical geometries and formulated with a variety of numerical procedures. The purpose of this test bed was to be able to evaluate combinations of preconditioners and solvers for applicability with different formulations (dense, sparse, hybrid, etc.) and different operators that yield differing conditions numbers. Information was also compiled for different sizes of systems to evaluate scalability issues. This information was organized in a web page under the Center for Computational Engineering to be used as a resource by the electromagnetics community at large. The URL for this information is http://cce.llnl.gov/solver. This is the fundamental mechanism that we used to transfer information among the collaborators on this effort.

Development of the solver framework

During the beginning of this activity, we began the development of the solver framework. The purpose of the solver framework is to 1) provide applications with a common interface to a variety of solvers and preconditioners, 2) provide developers with flexible and extensible system for incorporating new algorithms. Before we began to write code, we thought it would be wise to evaluate some existing linear solver packages. The phrase “six months in the laboratory can save you six hours in the library” is equally appropriate for software development. We evaluated PETSc (Portable Extensible Toolkit for Scientific Computing, Argonne), PMLP (Parallel Mathematical Library Project, Mississippi State), Hypre (LLNL), MTL (Matrix Template Library, University of Notre Dame), and ISIS++ (Iterative Scalable Implicit Solver in C++, Sandia National Laboratory). Of these solver libraries, the ISIS++ software seemed to have a simple and elegant design, and we chose to build upon the ISIS++ software rather than start entirely from scratch.

We re-wrote the ISIS++ software so that it could be used for complex-valued linear systems. In fact, we used generic programming techniques to re-write the software so the basic abstractions such a vectors, matrices, iterative solvers, and preconditioners are independent of data type. This way, the algorithms are independent of whether the underlying matrix entries are single precision real, double precision real, single precision complex, double precision complex, etc. We have generic (type-independent) MPI-based parallel versions of the following Krylov methods: Conjugate Gradient, Conjugate Gradient Squared, Bi-Conjugate Gradient Stabilized, Conjugate Gradient Normal Equations, Conjugate Gradient Normal Residual, Quasi-Minimal Residual, Generalized Minimal Residual, and Flexible Generalized Minimal Residual. We refer to our new framework as ISIS++2.0 and it is written in the C++ language. The image below depicts the structure of the framework.



The next step was to incorporate direct solvers into the framework. We required both a dense direct solver and a sparse direct solver. We evaluated PSPACES (University of Minnesota), SuperLU (University of California, Berkeley), OBLIO (Old Dominion University), and SPOOLES (DARPA/BOEING) as potential candidates for a sparse direct solver. The SPOOLES package was the only package that supported complex-valued matrices and was parallel. We completed the integration of the SPOOLES package into our solver framework in January of ‘01.

In addition, the ScaLapack library, which is the industry standard for dense systems, was incorporated into the framework.

Preconditioner research

We began early to investigate application specific preconditioners for dense complex-valued matrices resulting from integral-equation solution on Maxwell's equations. Initially, we concentrated on a class of preconditioners known as approximate inverse preconditioners. The original system is given by $\mathbf{Ax} = \mathbf{b}$. We first form a splitting $\mathbf{A} = \mathbf{A_1} + \mathbf{A_2}$ where $\mathbf{A_1}$ consists of a the very largest entries of $\mathbf{A}$, for example the largest 100 entries per row. The matrix $\mathbf{A_1}$ is used as the preconditioner, which requires the action of $\mathbf{A}_1{}^{-1}$. However since $\mathbf{A_1}$ is sparse we can use techniques commonly employed in finite difference and finite element applications to solve this problem. In the approximate inverse method, we compute a matrix $\mathbf{Q} \approx \mathbf{A}_1{}^{-1}$ either by minimizing the Frobenius norm or by employing so-called "mesh-neighbor" approaches. This approach is appealing since it is easy to parallelize and it works for complex-valued linear systems. In addition, it does not require symmetry or definiteness of the matrix. Our evidence indicates the approximate inverse preconditioner combined with an accelerator such as GMRES yields a method that is significantly more computationally efficient that the direct methods previously employed.

Shur Complement

One approach for improving the convergence of Krylov methods when applied to hybrid FEM-BEM matrices is to use the Schur Complement. We begin by writing the system of equations in block form

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

where A, B, and C are sparse blocks and D is a dense block. The subvector x_1 represents the internal unknowns and the subvector x_2 represents the boundary unknowns. The idea is that the matrix D is small enough to be "inverted" directly. Using surface-to-volume ratio arguments, the Schur Complement approach has a memory savings of $N^{2/3}$ compared to using a direct solver on the whole system. For example, for a computer with 8 GB of memory, the direct LU approach restricts us to problems of order 22k unknowns (complex double precision), whereas with the Schur Complement approach we can theoretically fit systems of order 2.89 million unknowns. The Schur Complement also permits the use FMM methods to effect the "inversion" of the D matrix.

Matrix based preconditioners

As mentioned earlier, we took a two-pronged approach to preconditioner research. The first tasks focused on matrix based preconditioners. The later tasks focused on direct operator preconditioners.

SPAI

We investigated Sparse Approximate Inverse methods as a possible preconditioner for our electromagnetics applications. Given a system A x = b the idea behind the SPAI is to construct a sparse matrix S^{-1} that somehow approximates A^{-1}. With CASC's SPAI algorithm the sparsity pattern of S^{-1} is equal to that of A, or of A*A, or of A*A*A, etc. The advantage of SPAI is that it is a general linear algebraic method that is not tuned to any particular differential operator. There were several research issues: first, CASC's SPAI code had to be modified to support complex arithmetic; second there was little theory describing convergence of SPAI preconditioned krylov methods so we had to resort to computational experiments to determine the efficiency of the method for our electromagnetics problems.

PILU

We also investigated Incomplete LU preconditioners. The original matrix A has a LU decomposition A = LU where L is lower triangular and U is upper triangular. The matrices L and U can be quite full even when A is sparse. The idea behind ILU is to compute sparse matrices L_s and U_s (i.e. incomplete) that approximate L and U, respectively. The parameter k denotes the amount of fill in L_s and U_s; typically there is an application dependent optimal value of k. ILU(k) has been a "workhorse" preconditioner in numerical PDE codes for many years, historically the primary issue was efficient parallelization of the ILU(k) method. Fortunately one of our academic collaborators had developed a nearly scalable parallel ILU(k) code. The next step was to develop a complex valued version of the ILU(k) and to try it on our electromagnetics problems.

Operator Normalization

The two primary cost factors associated with an iterative solver are the operation count required to compute a matrix-vector product, and the number of iterations required to achieve a given residual error level. A number of relatively mature algorithms are available which reduce the computational costs associated with the matrix-vector product operation. In contrast, well-conditioned BIEs are not presently available for most electromagnetic scattering problems. For this reason, the following discussion suggests a new framework for the development of effective and generalizable BIE renormalization/preconditioning methods. It is important to indicate that the proposed class of BIE renormalizations are also compatible with most of the hierarchical, multi-level schemes that reduce the computational costs of a matrix-vector product operation (such as FMM). Parallel implementations of the BIE renormalizations outlined below are also possible.

<u>BEM Preconditioners</u>

Preconditioners for boundary integral equations (BIEs) can be divided into two somewhat overlapping categories: those derived from matrix theory which are applied to the discretized form of the BIE operator, and those which are applied prior to discretization. The latter are typically based on the properties of the physical problem and/or BIE used to formulate an equivalent problem. The development of the former set of preconditioner for general linear matrix equations has a relatively long history and has resulted in a large number of methods designed for matrices having specific properties. However, experience indicates that the proper choice of such methods for a given scattering problem is often difficult.

For these reasons, we developed an alternative framework for the development of BIE preconditioner methods for electromagnetic scattering problems. This alternative approach is based both on a physical understanding of the actual scattering problem, and a physical interpretation of the BIEs used to formulate an equivalent (i.e., nonphysical) boundary value problem. This approach is distinct from most previous physically motivated BIE renormalization methods in that both the physical and nonphysical aspects of the equivalent boundary integral formulation are considered. Several investigators have considered physically-motivated renormalizations of the electric (EFIE), magnetic (MFIE) and combined (CFIE) field integral equations in order to facilitate the numerical simulation of specific scattering problems. However, the inability of these methods to adequately account for the differences between the physical and equivalent problems has made it difficult to extend their physical insights to more general scenarios.

The derived framework for preconditioner development is illustrated in Figure 9. As indicated in the figure, the proposed framework consists of a sequence of renormalizations. The focus of this effort was implementing this sequence and evaluating the performance on selected CEM solutions.

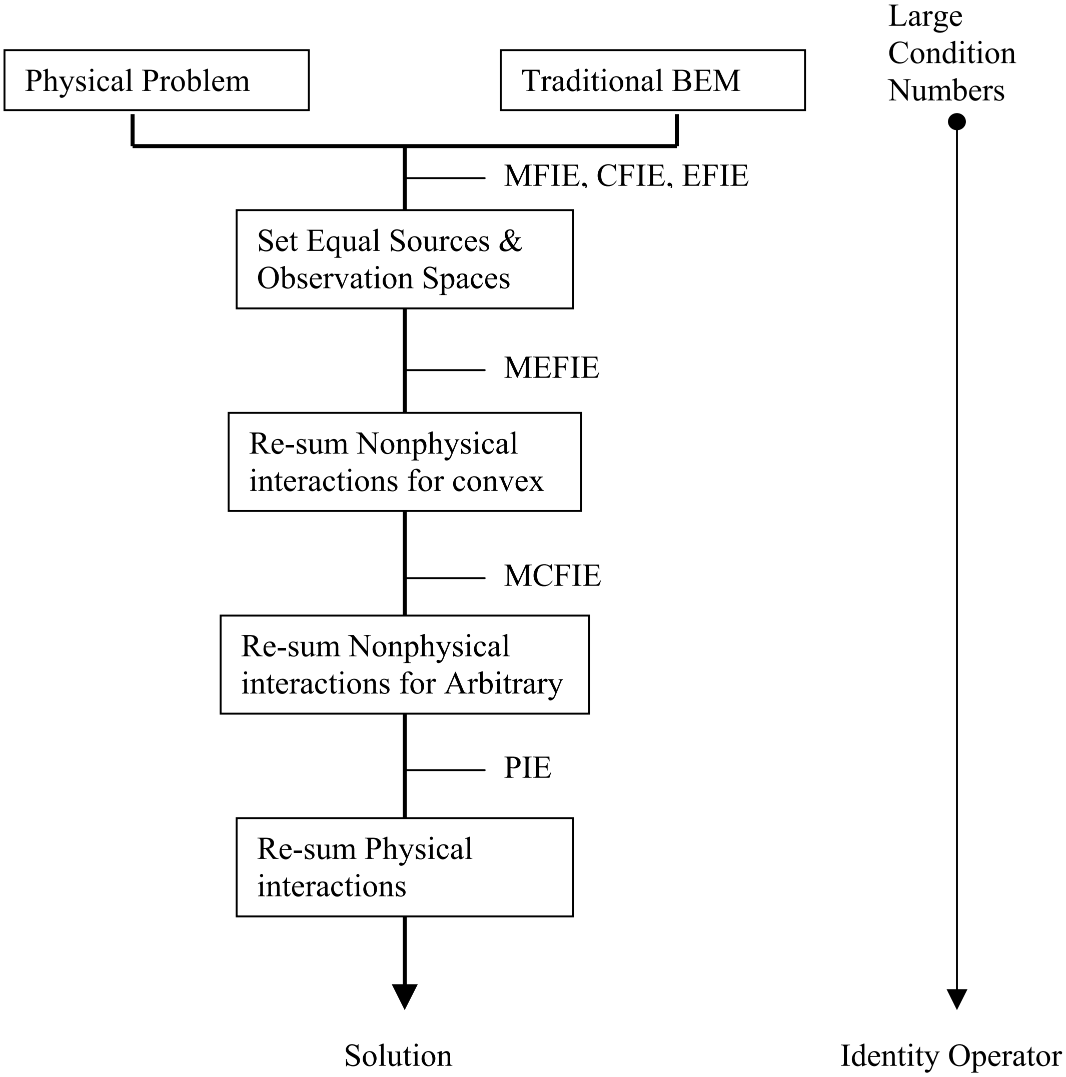

Large
Condition
Numbers
Physical Problem
Traditional BEM
MFIE, CFIE, EFIE
Set Equal Sources & Observation Spaces
MEFIE
Re-sum Nonphysical interactions for convex
MCFIE
Re-sum Nonphysical interactions for Arbitrary
PIE
Re-sum Physical interactions
Solution
Identity Operator

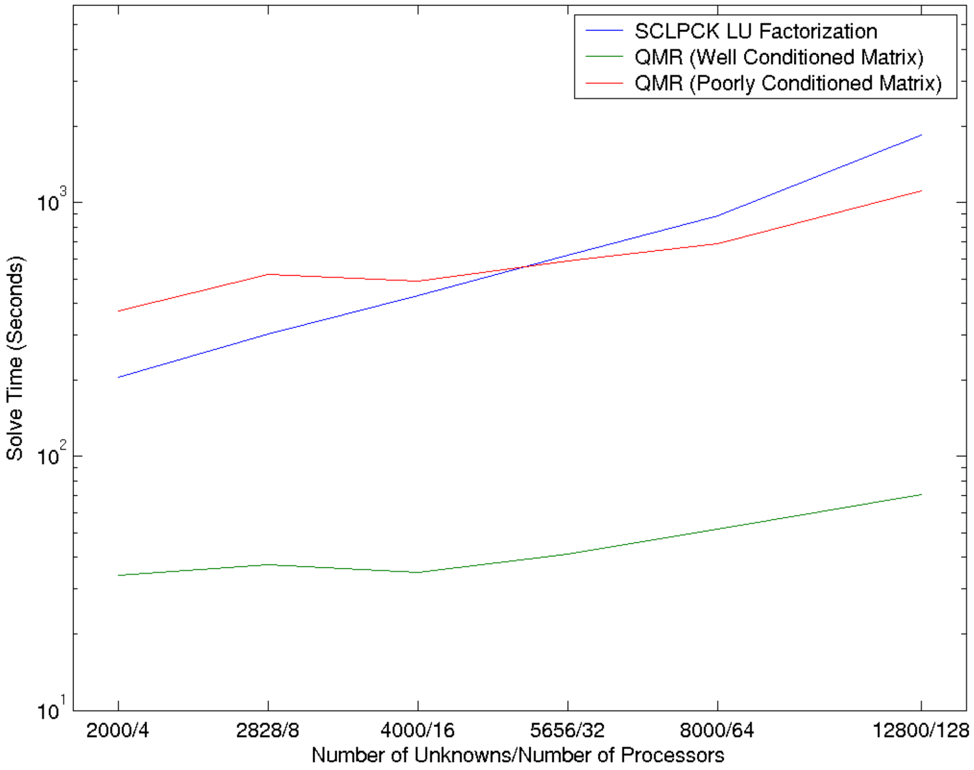
Figure 9. Derived framework for developing well-conditioned BIEs.

FMM

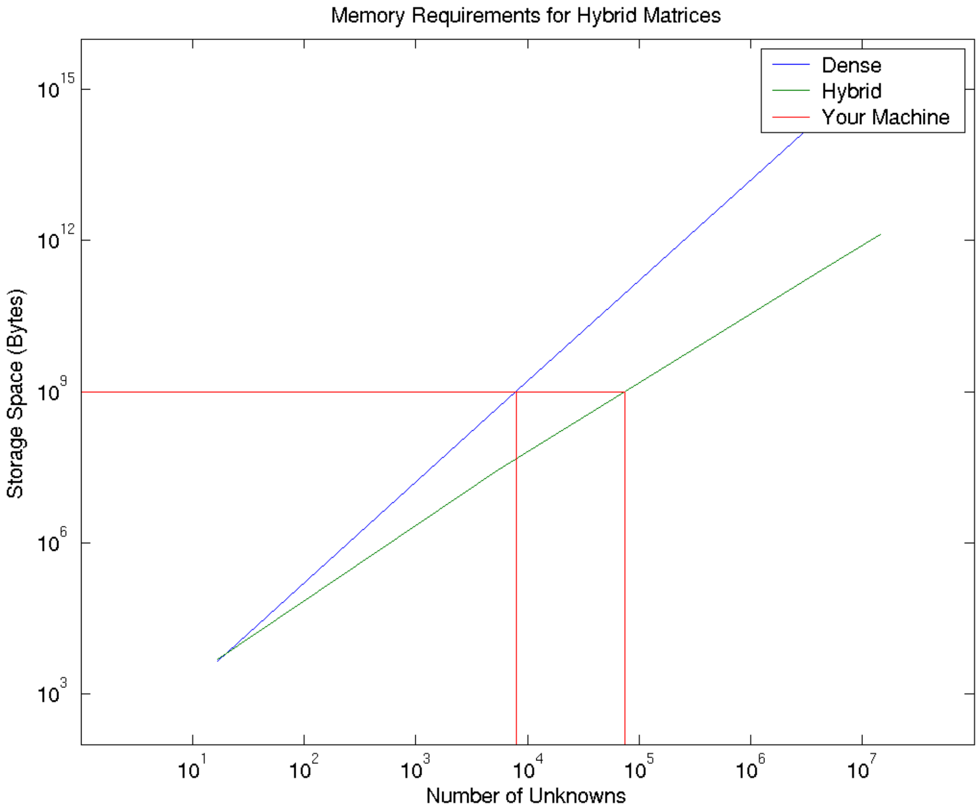
The fast multipole method (FMM) is a procedure for accelerating the "n body" interaction problem, which essentially describes most BEM solutions. Originally developed in astrophysics, FMM methods have recently been extended to CEM. These methods are used to accelerate the matrix-vector product associated with an iterative solution. We worked through the Institute for Scientific Computational Research (ISCR) in collaboration with Professor Karl Warnick of BYU. Dr. Warnick extended the basic FMM method to include the additional operators present in LLNL key CEM code suites.

Representative Results

In this section, representative results are given. The results shown in these figures were all obtained on Blue using a non-optimized version of ISIS2 compiled using KCC. Consequently they were intended to illustrate basic functionality and scalability characteristics of the ISIS2 library and not optimal performance characteristics.

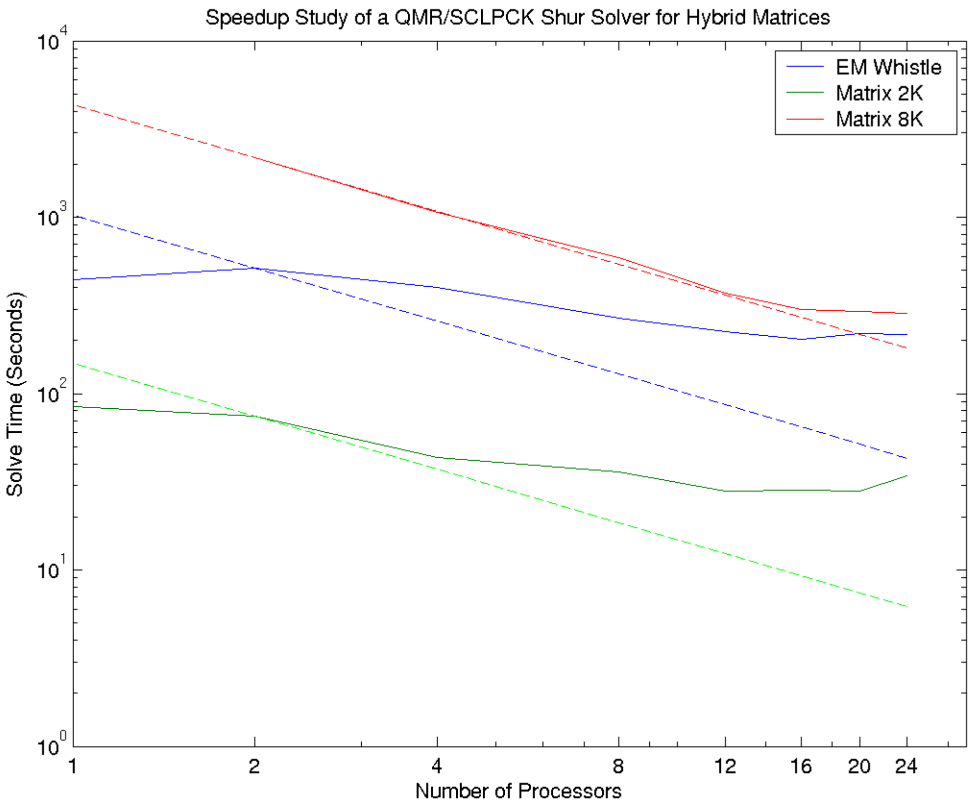


This figures shows a scalability study comparing the performance of ScaLAPACK's LU factorization with the Quasi Minimum Residual (QMR) solver in ISIS2 on dense, complex valued matrices. QMR was tested with two matrices: one well conditioned (cond(A) ~ log(N)), one poorly conditioned (cond(A) ~ N^2.) The number of unknowns (N) and the number of processors (P) were chosen such that N^2/P = 1x10^6

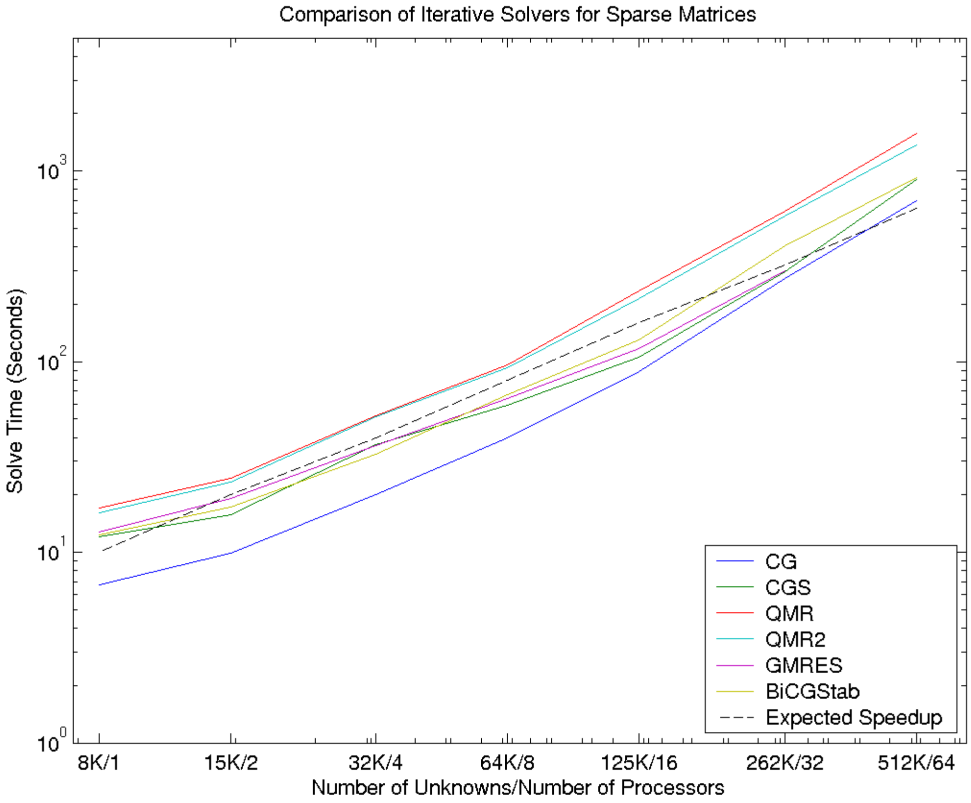


This figure approximates the increase in memory required to store a hybrid matrix using a standard dense scheme and our sparse/dense hybrid scheme as the number of unknowns grows. The red lines illustrate that by using this hybrid scheme we can increase the maximum problem size for a fixed memory footprint by a factor of ten.

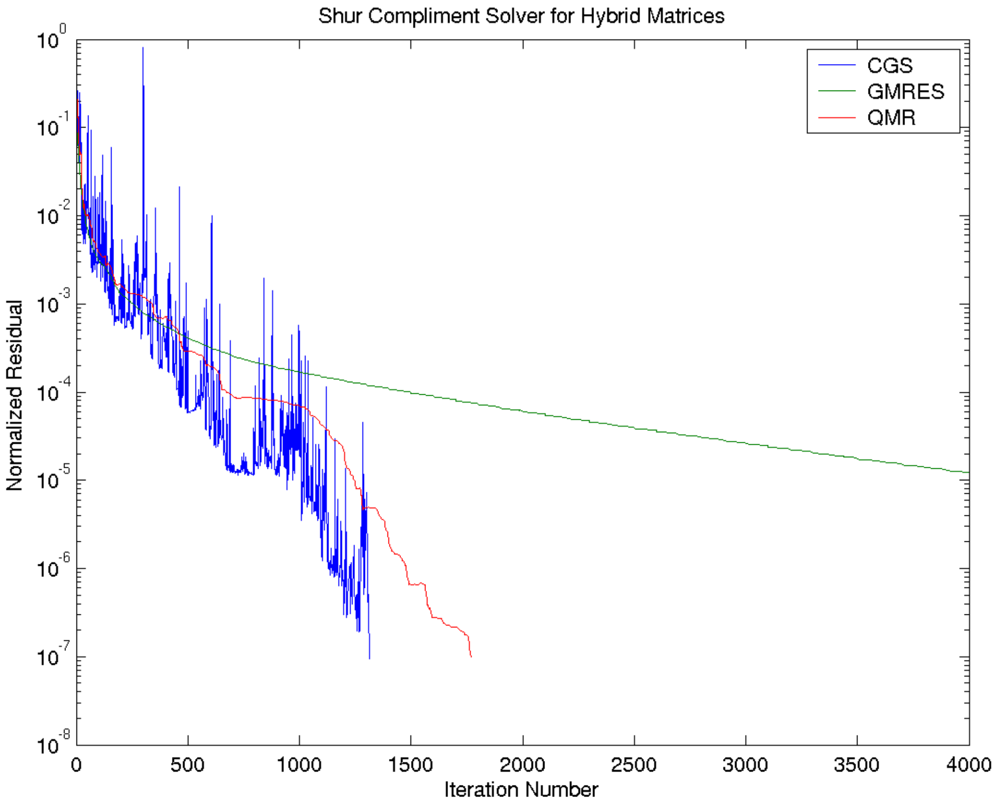
The approximations are based on the following hypothetical matrix with a dense portion made of the interactions of all of the nodes on the surface of a sphere and a sparse portion derived from the nearest neighbor (27 point) interactions of the nodes in the interior of the sphere.



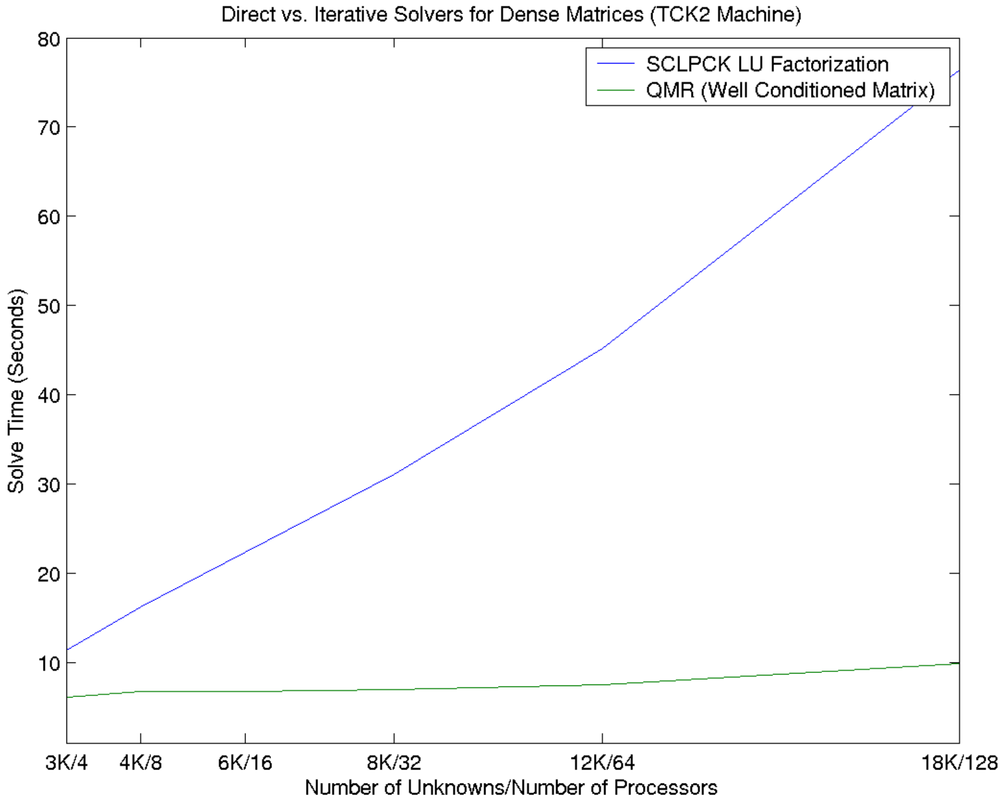
This shows the timing results for three hybrid problems as the number of processors is increased. “EM Whistle” has 3068 unknowns, “Matrix 2K” has 2010, and “Matrix 8K” has 7863. The dashed lines indicate what would be expected if we obtained “perfect speedup” i.e. twice as many processors should cut the solve time in half. The 8K example problem is the only problem that show considerable speedup indicating that this method shows its strength on larger problems.



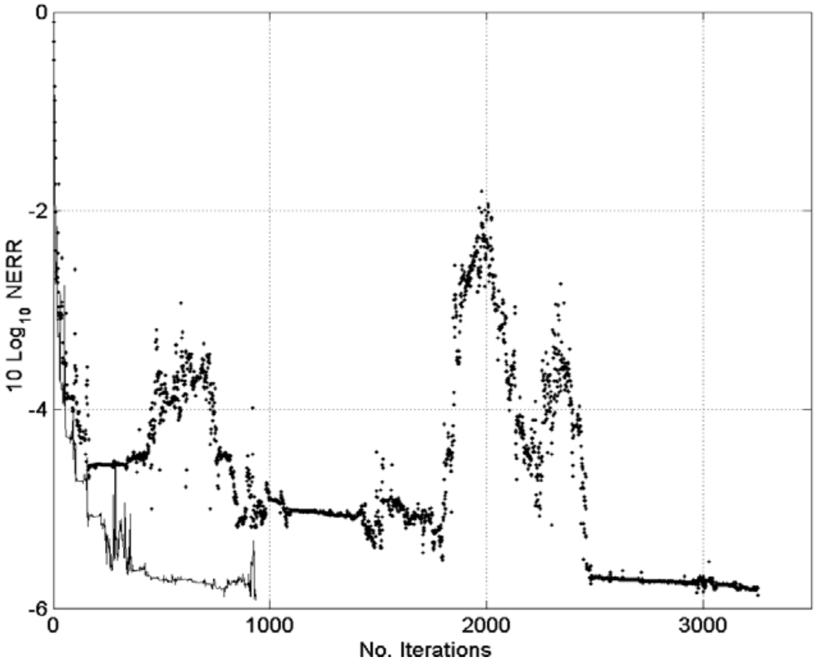
This figure is a sanity check of the iterative solvers in ISIS2. The matrices were Poisson 7 pt stencil, real valued. The right hand side and solution vectors were complex valued. The "Expected Speedup" curve assumes that the solve time will double when the number of unknowns and processors are both doubled.

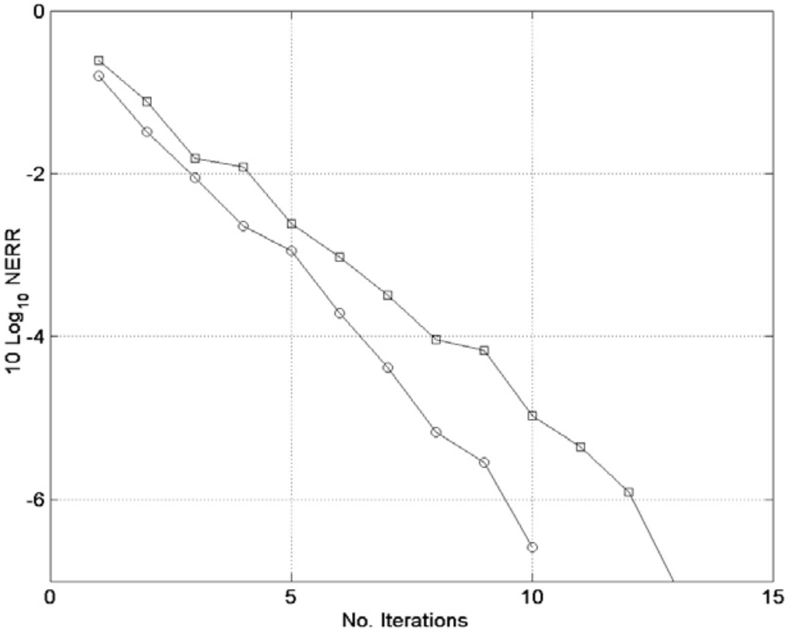


This figure shows the rate of convergence of the Shur Compliment solver using three different iterative solvers. The matrix in each case was the "EM_whistle" matrix. This is a very poorly conditioned complex valued matrix but the solvers work quite well.



This shows a scalability comparison of ScaLAPACK's LU factorization and QMR on a real valued matrix with condition number which grows like log(N). As expected, QMR's solve time remains roughly constant while LU grows nearly linearly with N. Note that N^2/NumProcs = const.





These two images display the convergence rate for scattering from λ/20 square PEC screen. BiCGStab convergence histories for standard electric field integral equation are shown in the first image and the convergence for the modified EFIE is shown in the later.
Number of basis elements: EFIE: N=408, 1045. MEFIE: N=408, 803.

References

[1] Sharpe R., Grant J., Champagne N., et. al., "EIGER: electromagnetic interactions generalized," UCRL-JC-126190. Also presented at the IEEE-APS symposium Quebec, Canada, July 13-18, 1997.

[2] Sharpe R., Grant J., Champagne N., et. al., "Electromagnetic interactions generalized (EIGER): algorithm abstraction and HPC implementation," UCRL-JC-130496. Also presented at 29th Plasma Dynamics and Laser Conference, Albuquerque, NM, June 15-18, 1998.

[3] Choi J., Demmel J., Dhillin I., Dongarra J., "ScaLAPACK – A portable linear algebra library for distributed memory computers," Computer Physics Communications, v. 97, n. 1, pp. 1 – 15, Aug. 1996.

[4] Cleary A., Falgout R., Henson V., et. al., "Robustness and scalability of algebraic multigrid," UCRL-JC-130718. Also submitted to SIAM J. Scientific Computing.

[5] Saad Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.

[6] Gupta A., Karypis G., Kumar V., "Highly Scalable Parallel Algorithms for Sparse Matrix Factorization," IEEE Trans. Parallel and Distributed Systems, v. 8, n. 5, pp. 502 – 520, 1997.

[7] Chow, E., "Scalable parallel computation of sparse approximate inverse preconditioners," UCRL-JC-131764. Also presented at SIAM Conference on Parallel Processing, San Antonio, TX, March 22 – 24, 1999.

[8] Greengard L, Huang J., Rokhlin V., Wandzura, S., "Accelerating fast multi-pole methods for the Helmholtz equation at low frequencies," IEEE Computational Science and Engineering, v. 5, n. 3, July, 1998.

[9]R.J.Adams, A Class of Robust and Efficient Iterative Methods for Wave Scattering Problems, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, December 1998.

[10]R.J.Adams and G.S.Brown, Stabilization procedure for electric .eld integral equation, Electronics Letters, 35(23):2015 –2016,1999.

[11]R.J.Adams, Physical and analytical properties of a stabilized electric field integral equation, IEEE Transactions on Antennas and Propagation .Submitted February 2001.

[12]R.J.Adams, A numerical implementation of the modified electric field integral equation, IEEE Transactions on Antennas and Propagation, In preparation.

[13]R.J.Adams, Combined field integral equation formulations for electromagnetic scattering from convex geometries, IEEE Transactions on Antennas and Propagation, Submitted October,2000.

[14]R.J.Adams, On the development of a physical boundary integral equation for electromagnetic scattering problems, IEEE Transactions on Antennas and Propagation, Submitted October, 2000.

[15]G.C.Hsiao and R.E.Kleinman, Mathematical foundations for error estimation in numerical solutions of integral equations in electromagnetic, IEEE Transactions on Antennas and Propagation ,45(3):316 –328,1997.

[16]H.Contopanagos, B.Dembart, M.Epton, J.J.Ottusch, V.Rokhlin, J.Visher, and S.Wandzura, Well-conditioned boundary integral equations for three-dimensional electromagnetic scattering, Technical report,Yale,2000.